

Appl. No. 10/013,097
Reply Filed: November 17, 2004
Reply to Office Action of May 17, 2004

AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph at Page 4, lines 14-18, and starting with "In the first case," with the following amended paragraph:

In the first case, in which an attribute is added in a new implementation, an application supports backward compatibility to allow the expected attribute not to be present in data received from another application. Because the added attribute is not present in old files, a mismatch is detected. The mismatch can be corrected by synthesizing the attribute or by notifying a new implementation that includes code to convert old objects into new ones.

Please replace the paragraph at Page 4, line 29 to Page 5, line 7, and starting with "In the fourth case," with the following amended paragraph:

In the fourth case, in which an attribute is removed in a new implementation, an application supports forward compatibility to allow an old application to accept an object that does not include an attribute that is expected but not present. There are several options for handling this case. The attribute may be synthesized and a default value may be supplied, e.g., integer = 0, real = 0.0, string = "". Alternatively, the attribute may be synthesized and a default value may be obtained from a value or rule built ~~in to~~ into the implementation or from a value or rule stored in the file along with the data, for example, by adding default values to an enhanced AAF dictionary. Alternatively, a new value for the attribute may be computed using plug-in code that is specifically for converting new objects into old ones and may be stored or described in the file along with the data.

Please replace the paragraph at Page 5, line 27 to Page 6, line 17, and starting with "AAF defines," with the following amended paragraph:

AAF defines a packaging format for data. This packaging format includes metadata 100, "internal" essence 102 (for simple essence such as graphics) and references 104 to "external" essence 106 (for complex essence such as video data). A dictionary 108 is a description of the

Appl. No. 10/013,097
Reply Filed: November 17, 2004
Reply to Office Action of May 17, 2004

schema to which the metadata conforms. Files including data in an AAF format include this dictionary. A "plug-in" mechanism 110 specifies any plug-in used by the data, for example, a codec or other executable code for a particular kind of data. The metadata 100 specifies one or more content packages 112, each including one or more content items 114 and/or one or more other content packages 112. One of the content items is an index 116 of the data in the package. An item may refer to one or more objects. Each object has one or more attributes (also called properties), that are defined by a key (or name), a value, and optionally a length or other information descriptive of the value. Further details about the AAF specification may be found at <http://www.aafassociation.org> ~~http://www.aafassociation.org~~.

Please replace the paragraph at Page 6, lines 11-18, and starting with "How AAF is used," with the following amended paragraph:

How AAF is used by an application will now be described in connection with Fig. 2. One or more client applications 200 access data in AAF format stored on a storage system 202 via an AAF software development kit (SDK) 204. The SDK receives commands from an application 200 through a public API 206, and issues commands to an operating system 208 through an OS API 210. A media engine 212 communicates with the storage system 202 and client application 200, as well as an object manager 214 of the AAF SDK to process and/or present media data. The AAF SDK currently is publicly available through <http://www.aafassociation.org> ~~http://www.aafasoeiation.org~~.

Please replace the paragraph at Page 6, lines 19-30, and starting with "The AAF SDK includes," with the following amended paragraph:

The AAF SDK includes utilities 216 useful for developers and conversion applications 218 for converting data between different metadata schemas. Such differences might exist between the schema of the data being accessed and the schema of data used by external applications. The utilities and conversion applications may be "plug-in"s that may be added to the AAF SDK through a plug-in mechanism provided by the AAF SDK. An API broker 220 implements the public API 206 that is used by the utilities 216, conversion applications 218 and client

Appl. No. 10/013,097
Reply Filed: November 17, 2004
Reply to Office Action of May 17, 2004

applications 200 to access the data model 222, which in turn accesses the object model 214 to access stored data. The data model 222 includes a mechanism to process objects (attribute sets) and modify them. The mechanisms for maintaining interoperability with different metadata schemas may be added to this data model. The object manager 214 includes a mechanism to handle constrained evolution, as described below. The object manager 214 implements an API 224 that allows access by both the data model 222 and the media engine 212.

Please replace the paragraph at Page 10, lines 4-9, and starting with "More particularly," with the following amended paragraph:

More particularly, for backward compatibility, if a new implementation opens a file from an old implementation, it registers ~~EvolvedPropertyDefs~~ EPDs for all the old properties it does not understand. In each of these EPDs the Target name is the Key of the attribute in the new schema, and the Source name is the Key of the attribute in the old schema. These EPDs may be created from scratch by the newer application, or may be read in from an auxiliary file that includes these EPDs. This auxiliary file may contain only the necessary EPDs.